CS 473 ✧ Fall 2024
Conflict Midterm 2 Problem 1 Solution

Suppose we initialize an array $A[1..n]$ by setting $A[i] = i$ for all i, and then randomly shuffle the array. After shuffling, each of the $n!$ possible permutations of the array A is equally likely. An index i is called a ***fixed point*** of the shuffling permutation if $A[i] = i$. Let X denote the number of fixed points of this random permutation.

(a) What is the exact value of $\mathrm{E}[X]$?

(b) What is the exact value of $\mathrm{E}[X^2]$?

Prove that both of your answers are correct.

Solution:

(a) For each index i, define an indicator variable X_i that is equal to 1 if and only if $A[i] = i$. Then $X = \sum_i X_i$. Each element $A[i]$ is equally likely to have any value from 1 to n, so $\Pr[A[i] = i] = 1/n$. So linearity of expectation implies

$$\mathrm{E}[X] \;=\; \sum_{i=1}^{n} \Pr[X_i = 1] \;=\; \sum_{i=1}^{n} \Pr[A[i] = i] \;=\; \sum_{i=1}^{n} \frac{1}{n} \;=\; \boxed{1}$$

(b) First we observe that

$$\begin{aligned} X^2 \;&=\; \left(\sum_{i=1}^{n} X_i\right)^2 \;=\; \left(\sum_{i=1}^{n} X_i\right)\left(\sum_{j=1}^{n} X_j\right) \\ &=\; \sum_{i=1}^{n} X_i^2 + \sum_{i \neq j} X_i X_j \\ &=\; \sum_{i=1}^{n} X_i + \sum_{i \neq j} X_i X_j \;=\; X + \sum_{i \neq j} X_i X_j \end{aligned}$$

So linearity of expectation and part (a) imply

$$\mathrm{E}[X^2] \;=\; 1 + \sum_{i \neq j} \Pr[X_i X_j = 1]$$

We have $X_i X_j = 1$ if and only if both $A[i] = i$ and $A[j] = 1$. $A[i]$ is equally likely to have any value from 1 to n; once $A[i]$ is fixed, $A[j]$ is equally likely to have any value *except* $A[i]$. Thus, $\Pr[X_i X_j = 1] = 1/n(n-1)$. We conclude

$$\mathrm{E}[X^2] \;=\; 1 + \sum_{i \neq j} \frac{1}{n(n-1)} \;=\; 1 + n(n-1) \cdot \frac{1}{n(n-1)} \;=\; 1 + 1 \;=\; \boxed{2}$$

■

Rubric: 10 points = 5 for part (a) + 5 for part (b). This is more detail than necessary for full credit. These are not the only correct proofs.

CS 473 ✧ Fall 2024
Conflict Midterm 2 Problem 2 Solution

Let P be a multilinear polynomial with n variables $x_1, \ldots, x_n$ and m terms, where the degree of every term is exactly $n/10$. We call a set of variables $H \subseteq \{x_1, \ldots, x_n\}$ a ***hitting set*** for P if H contains at least one variable from each term of P. Suppose we randomly generate H by independently including each variable x_i with probability $p = (c \log m)/n$, for some constant $c \geq 1000$.

(a) ***Prove*** that for any fixed term in P, the set H contains at least one variable from that term with probability at least $1 - 1/m^3$.

(b) ***Prove*** that H contains at most $c^2 \log m$ variables with probability at least $1 - 1/m^2$.

(c) ***Prove*** that H is a hitting set of size $O(\log m)$ with probability at least $1 - 1/m$.

Solution: Let's assume that log means $\ln = \log_e$.

(a) Let T be any single term in P. Because each variable in T is independently excluded from H with probability $1 - p$, we have

$$\Pr[H \text{ contains no variable in } T] \;=\; (1-p)^{n/10} \;=\; \left(1 - \frac{c \log m}{n}\right)^{n/10} \;\leq\; e^{-(c \log m)/10}$$

by The World's Most Useful Inequality. Setting $\boldsymbol{c = 1000}$ gives us

$$\Pr[H \text{ contains no variable in } T] \;\leq\; e^{-100 \log m} \;=\; \frac{1}{m^{100}} \;\leq\; \frac{1}{m^3}.$$

(b) We immediately have

$$\mathrm{E}[|H|] \;=\; pn \;=\; c \log m,$$

and therefore

$$\Pr\Big[|H| > c^2 \log m\Big] \;=\; \Pr\Big[|H| > c \cdot \mathrm{E}[|X|]\Big]$$

Because $|H|$ is a sum of independent indicator variables, one for each element of U, the Chernoff bound $\Pr[|X| > (1+\delta)\mu] < e^{-\delta\mu/2}$ with $\delta = c - 1$ implies

$$\Pr\Big[|H| > c \cdot \mathrm{E}[|X|]\Big] \;\leq\; e^{-((c-1)c \log m)/2}$$

Finally, setting $\boldsymbol{c = 1000}$ gives us

$$e^{-((c-1)c \log m)/2} \;\leq\; e^{-499500 \log m} \;=\; \frac{1}{m^{499500}} \;\leq\; \frac{1}{m^2}.$$

(c) H is ***not*** a hitting set of size at most $c^2 \log n$ if and only if either (1) H excludes all variables in one of the m terms of P, or (2) $|H| > c^2 \log m$. Parts (a) and (b) and the union bound imply that H is ***not*** a hitting set of size at most $c^2 \log n$ with probability at most

$$\frac{m}{m^3} + \frac{1}{m^2} = \frac{2}{m^2} \le \frac{1}{m}.$$

(The last inequality breaks down when $m = 1$, but in that case the probability is *trivially* at most $1/m = 1$.) ■

Rubric: 10 points = 4 for part (a) + 4 for part (b) + 2 for part (c). This is more detail than necessary for full credit.

CS 473 ✧ Fall 2024
Conflict Midterm 2 Problem 3 Solution

Suppose you have a set X of n items from some universe $\mathcal{U}$ that you want to store in a simple array $A[1\,..\,n]$. Your manager has given you five *access* functions h_1, h_2, h_3, h_4, h_5, each of which takes an element of $\mathcal{U}$ as input and returns an integer between 1 and n as output, in constant time.

These access functions are ***flawless*** if it is possible store each element $x \in X$ at one of the five addresses $A[h_1(x)]$, $A[h_2(x)]$, $A[h_3(x)]$, $A[h_4(x)]$, or $A[h_5(x)]$, with no collisions—each array entry $A[i]$ must store exactly one element of X.

Describe and analyze an algorithm to determine whether the given access functions are flawless. The input to your algorithm is the set X and the access functions h_1, h_2, h_3, h_4, h_5.

Solution: Suppose $X = \{x_1, x_2, \ldots, x_n\}$. Define a bipartite graph $G = (L \sqcup R, E)$ with vertices $L = X$ and $R = [n]$, which contains an edge between $x_i \in L$ and $j \in R$ if and only if $h_1(x_i) = j$ or $h_2(x_i) = j$ or $h_3(x_i) = j$ or $h_4(x_i) = j$ or $h_5(x_i) = j$. The graph G has exactly $2n$ vertices and at most $5n$ edges.

Compute a maximum matching M in this graph, as described in class, and then report that the given access functions are flawless if and only if M contains exactly n edges. The algorithm runs in $O(VE) = \boldsymbol{O(n^2)}$ ***time***. ■

Rubric: 10 points: standard graph reduction rubric

CS 473 ✧ Fall 2024

Conflict Midterm 2 Problem 4 Solution

Suppose you are given a 3CNF formula with n variables $x_1, \ldots, x_n$ and m clauses, where $m \geq 100$.

(a) Suppose we independently assign each variable x_i to be TRUE or FALSE with equal probability. What is the exact expected number of ***unsatisfied*** clauses under this assignment?

(b) ***Prove*** that the probability that at least $m/8+1$ clauses are ***unsatisfied*** is at most $1 - C/m$ for some constant C.

(c) Part (b) implies that under a random assignment, the number of ***satisfied*** clauses is at least $7m/8$ with probability at least C/m. Using this fact, describe an *efficient* randomized algorithm that *always* finds an assignment that ***satisfies*** at least $7m/8$ clauses, and analyze its expected running time.

Solution: Suppose we are given a 3CNF formula $\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, where each C_j is a clause with three literals.

(a) For each index $1 \leq j \leq m$, let $X_j = 1$ if the random assignment does ***not*** satisfy the clause C_j and $X_j = 0$ otherwise. Each clause C_j if and only if all its literals are FALSE. Since each literal is TRUE or FALSE with equal probability, we have $\Pr[X_j = 1] = 1/8$.

Let $X = \sum_{j=1}^m X_j$ denote the number of ***unsatisfied*** clauses. The *expected* number of unsatisfied clauses is exactly

$$\mathrm{E}[X] = \sum_{j=1}^{m} \Pr[X_j = 1] = \boxed{\frac{m}{8}}$$

(b) Markov's inequality immediately implies

$$\begin{aligned}
\Pr[X \geq m/8 + 1] &\leq \frac{\mathrm{E}[X]}{m/8+1} \\
&= \frac{m/16}{m/8+1} && \text{[part (a)]} \\
&= 1 - \frac{1}{m/8+1} && \text{[math]} \\
&\leq 1 - \frac{1}{m/4} && [\tfrac{1}{1+t} \geq \tfrac{1}{2t}] \\
&= \boxed{1 - \frac{4}{m}} && \text{[math]}
\end{aligned}$$

(c) Our algorithm repeatedly generates and tests independent random assignments, until we find one that satisfies at least $7m/8$ clauses.

Each iteration of the algorithm takes $O(m+n)$ time to generate a random assignment and then check each clause by brute force. Part (b) implies that in each iteration, we generate a good assignment with probability at least $4/m$. Let Y denote the number of iterations required to find a good assignment; we have

$$\mathrm{E}[Y] \le 1 + (1 - 4/m)\,\mathrm{E}[Y] \implies \mathrm{E}[Y] = m/4.$$

Thus, our brute-force algorithm runs in ***$O(m(m+n))$ expected time***. ■

Rubric: 10 points = 3 for part (a) + 3 for part (b) + 4 for part (c)